**User Manual**

**Smart Color Sorting System with ESP8266**

**This exercise has 3 modules of operations namely.,**

* + **Module I – Programming**
  + **Module II – Hardware Configuration**
  + **Module III- Compilation and Execution**

**Let us see the step-by-step operations of the modules**

**Module -I Programming**

Step 1: Install and Open Arduino IDE

Step 2: After opening the IDE, this window will appear. Start writing code in the provided space (highlighted space) and save the code

A screen shot of a computer

Description automatically generated

Steps 3: #include <Servo.h>

#include <ESP8266WiFi.h>

#include <ThingSpeak.h>

const char \* myWriteAPIKey = "";

unsigned long myChannelNumber = 3242;

const char \*ssid = ""; // Enter your WiFi Name

const char \*pass = ""; // Enter your WiFi Password

WiFiClient client;

Servo pickServo;

Servo dropServo;

const int s0 = 02; //D4;

const int s1 = 14; //D5;

const int s2 = 12; //D6;

const int s3 = 13; //D7;

const int out = 15; //D8;

int red = 0;

int green = 0;

int blue = 0;

int redcolor = 0;

int orangecolor = 0;

int greencolor = 0;

int yellowcolor = 0;

int CLOSE\_ANGLE = 30; // The closing angle of the servo motor arm

int OPEN\_ANGLE = 10; // The opening angle of the servo motor arm

void setup() {

Serial.begin(9600);

pinMode(s0, OUTPUT);

pinMode(s1, OUTPUT);

pinMode(s2, OUTPUT);

pinMode(s3, OUTPUT);

pinMode(out, INPUT);

digitalWrite(s0, HIGH);

digitalWrite(s1, HIGH);

pickServo.attach(04);//D2

dropServo.attach(00); //D3

pickServo.write(CLOSE\_ANGLE); // Initialize the pickServo to closed position

dropServo.write(73); // Initialize the dropServo to default position

ThingSpeak.begin(client);

Serial.println("Connecting to WiFi...");

WiFi.begin(ssid, pass);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

}

void loop() {

digitalWrite(s2, LOW);

digitalWrite(s3, LOW);

**// Measure color intensity for red, green, and blue**

red = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);

digitalWrite(s3, HIGH);

blue = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);

digitalWrite(s2, HIGH);

green = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);

Serial.print("R Intensity: ");

Serial.print(red);

Serial.print(" G Intensity: ");

Serial.print(green);

Serial.print(" B Intensity: ");

Serial.println(blue);

**// Color detection based on intensity thresholds**

if (red > 29 && red < 39 && green > 83 && green < 93 && blue > 69 && blue < 78) {

// Red detected

dropAndRecordColor(73, &redcolor, 1);

}

else if (green > 65 && green < 75 && blue > 60 && blue < 68) {

// Orange detected

dropAndRecordColor(107, &orangecolor, 2);

}

else if (red > 36 && red < 46 && green > 37 && green < 46) {

// Green detected

dropAndRecordColor(132, &greencolor, 3);

}

else if (red > 25 && red < 34 && green > 28 && green < 37 && blue > 43 && blue < 53) {

**// Yellow detected**

dropAndRecordColor(162, &yellowcolor, 4);

}

delay(1000);

}

void dropAndRecordColor(int servoAngle, int\* colorCount, int fieldNumber) {

dropServo.write(servoAngle); // Drop the item

delay(700); // Wait for the item to drop

openServo(); // Open trap door to release the item

delay(200);

closeServo(); // Close the trap door

**// Increment the count for the detected color**

(\*colorCount)++;

**// Update ThingSpeak field with the color count**

ThingSpeak.writeField(myChannelNumber, fieldNumber, \*colorCount, myWriteAPIKey);

}

void openServo() {

pickServo.write(OPEN\_ANGLE); // Open the trap door

}

void closeServo() {

pickServo.write(CLOSE\_ANGLE); // Close the trap door

}

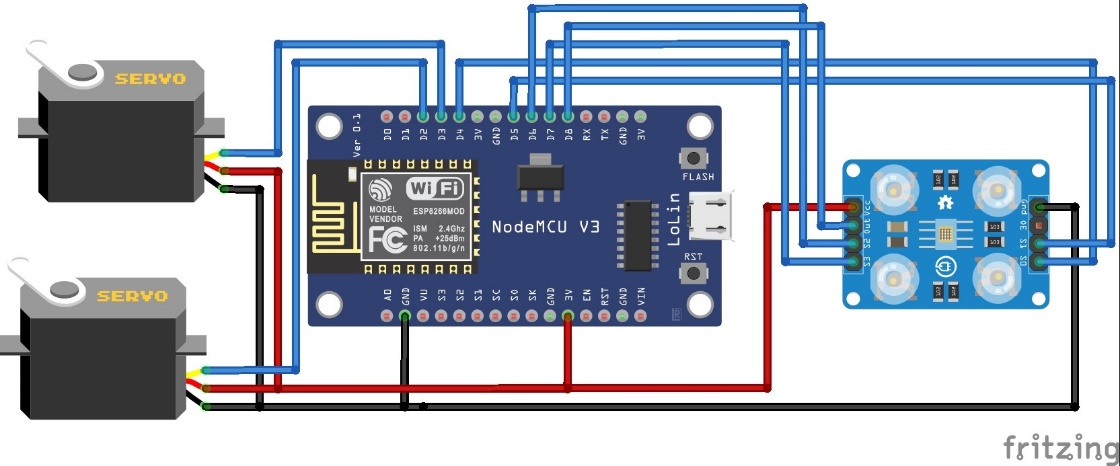
**Module -II Hardware Configuration**

**Step 1:**

* + We have the following components with us.
    - ESP8266
    - TCS3200 Color Sensor
    - 2 Servo Motors

**Step 2: Let us connect TCS3200 Color Sensor and Servo Motors with ESP8266 pins.**

* + From 8 pins of TCS3200 we must connect 7 pins, they are VCC, GND, Out, S0, S1, S2, and S3
  + Connect the VCC of TCS3200 to the 3V of ESP8266
  + Next, connect the GND of TCS3200 to the GND of ESP8266
  + Next, connect the S0 pin of TCS3200 to the D4 of ESP8266
  + Next, connect the S1 pin of TCS3200 to the D5 of ESP8266
  + Next, connect the S2 pin of TCS3200 to the D6 of ESP8266
  + Next, connect the S3 pin of TCS3200 to the D7 of ESP8266
  + Next, connect both Servo Motors GND to the GND of ESP8266
  + Next, connect both Servo Motors VCC to the 3V of ESP8266
  + Next, connect one Servo Motor’s control signal to the D2 and second Servo Motor’s control signal to the D3 of ESP8266



**Step 3:**

* After connecting the sensor to the controller i.e. **ESP8266** now connect your **ESP8266** board to your computer using a micro-USB cable.

**Module -II Executing the code**

Step 1: In Arduino IDE, check the port and board status.

1.1 Click on Tools -> Port -> Select the COM port (where USB is connected).

1.2 Click on Tools -> Board -> Select the board

A screenshot of a computer

Description automatically generated

Step 2: After selecting the port and board, compile and upload the code

A blue and black rectangle with white text

Description automatically generated

Icon for Compilation

A blue and black rectangle with white text

Description automatically generated

Icon for Uploading the code

Step 3: After uploading is complete, click on the ‘Serial Monitor’ for the output

A blue and black rectangle with white text

Description automatically generated

Click on the icon to open serial monitor

**For ThingSpeak:**

1. Open a web browser and go to the ThingSpeak website by visiting <https://thingspeak.com/>.
2. On the ThingSpeak homepage, look for the "Sign Up" button, usually located in the top right corner of the page. Click on it.
3. As a new user, Click on Create one.
4. Provide the necessary credentials and create your thing speak account.
5. After Logging in you will be able to view the below shown page for creating a channel which is to interface the hardware to Cloud using API key.
6. Click New Channel to create a channel for your application as shown below, here for an Example, Flood monitoring with Two fields has been chosen and scroll down and click “Save Channel”.
7. After saving the channel, you can see the Channel Id and Other data has been Displayed and from the below shown menu choose API keys.
8. In API keys, Write API and Read API key will generate, Use the key in the Arduino Embedded C Code to Fetch the data in Cloud.